

UNITED STATES PATENT APPLICATION

**MULTIPLE CHANNEL INTERFACE FOR COMMUNICATIONS
BETWEEN DEVICES**

INVENTORS

BRIAN B. MEARS

of Tempe, AZ, USA

MARK N. FULLERTON

of Austin, TX, USA

NICHOLAS J. KOHOUT

of Austin, TX, USA

Schwegman, Lundberg, Woessner, & Kluth, P.A.
1600 TCF Tower
121 South Eighth Street
Minneapolis, Minnesota 55402
ATTORNEY DOCKET 884.481US1
Client Ref. No. P11715

MULTI-CHANNEL INTERFACE FOR COMMUNICATIONS BETWEEN DEVICES

5

Field of the Invention

The present invention relates generally to communications between devices, and more particularly to a multi-channel interface for communications between devices, circuits, semiconductor chips or the like.

10

Background Information

Electronic systems and devices are being required to perform more functions in shorter periods of time. Such electronic systems and devices contain multiple semiconductor chips, circuits and or the like. The semiconductor chips or circuits are typically required to communicate with each other in order to perform particular operations or functions. To accomplish these communications, multiple communications links or conductors are required to interconnect the semiconductor chips or circuits. These electrical connections can occupy considerable area on a substrate and can also require multiple pins on each of the chips for the inter-chip communications. Complex software may also be needed to implement the inter-chip or circuit communications and to accurately direct or address data signals or information to various components to perform the particular functions or operations.

15

20

Accordingly, for the reason stated above, and for other reasons that will become apparent upon reading and understanding the present specification, there is a need for a multiple channel communications interface that minimizes the number of inter-chip connects and pins on each chip. Additionally, there is a need for a multiple channel communications interface that simplifies the software required for implementation and minimizes overhead, and is scalable to provide more or fewer communications channels depending upon design constraints.

25

Brief Description of the Drawings

Figure 1 is a schematic block diagram of an electronic systems in accordance the present invention.

5 Figure 2 is a schematic block diagram of another electronic system in accordance with the present invention.

Figure 3 is a schematic block diagram of a communications interface in accordance with an embodiment of the present invention.

Figure 4 is a schematic block diagram of a communications interface in accordance with another embodiment of the present invention.

10 Figure 5 is a schematic block diagram of a communications interface illustrating examples of control registers that may be used to transmit data to other chips or devices in accordance with the present invention.

Figure 6 is a table showing an example of a bit layout and bit definitions for a channel status register in accordance with the present invention.

15 Figure 7 is a table showing an example of a bit layout and bit definitions for a channel configuration register in accordance with the present invention.

Figure 8 is a table showing an example of a bit layout and bit definitions for an interface interrupt identification register in accordance with the present invention.

20 Figure 9 is a schematic block diagram of a communications interface illustrating examples of control registers that may be used to receive data from other chips or devices in accordance with the present invention.

Figure 10 is a schematic block diagram of a communications interface illustrating examples of control registers that may be used to send and receive general purpose input/output (GPIO) signals or data in accordance with the present invention.

25 Figure 11 is a block schematic diagram of a source communications interface and a target communications interface coupled by different communications links or pins in accordance with the present invention.

Figure 12 is a table illustrating an example of channel assignments for a communications interface in accordance with the present invention.

Figure 13 is an example of signal waveforms to transmit a message or data between a source communications interface and a target communications interface in accordance with the present invention.

Figure 14 is an example of signal waveforms to select a new channel to transmit data between a source communications interface and a target communications interface in accordance with the present invention.

Figure 15 is an example of signal waveforms illustrating message flow control in accordance with the present invention.

Figure 16 is a flowchart of an example of a method for transmitting data between semiconductor chips or devices in accordance with the present invention.

Description of the Preferred Embodiments

In the following detailed description of the preferred embodiments, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

Figure 1 is a schematic block diagram of an electronic system 100 in accordance with an embodiment of the present invention. The electronic system 100 includes at least two semiconductor chips 102a and 102b, devices or circuits that communicate with one another. The electronic system 100 may include additional semiconductor chips, devices or circuits 102c and 102d. The electronic system 100 may also be formed as a single chip and the circuits or devices 102 may each be an on-chip silicon module. For example, semiconductor chips 102a and 102c may be processors, such as central processing units (CPUs), digital signal processors (DSPs) for the like, and the semiconductor chips 102b and 102d may be memory devices, peripheral equipment for the like. The chips 102a and 102c are coupled to an internal bus 110, such as a processor bus or a peripheral bus, and the internal bus 110 is coupled to a first communications interface 112 or multiple channel interface for communications between the chips 102a, 102b, 102c and 102d. The first communications interface 112

may be a communications interface between any different types of chips or may be a broadband-to-multimedia (BB-MM) interface for communications between a multimedia processor and a baseband chip. The first communications interface 112 is electrically coupled to a second communications interface 114 by a plurality of outgoing or outbound links or pin connections 116 and a plurality of incoming or inbound links or pin connections 118. The outbound links or pins 116 from the first communications interface 112 are the inbound links to the second communications interface 114 and the inbound links 118 to the first communications interface 112 are the outbound links from the second communications interface 114. The second communications interface 114 is electrically coupled to an internal bus 120 and the bus 120 is electrically coupled to be semiconductor chips 102b and 102d, if there are more than one chip 102 coupled to the second communications interface 114.

Figure 2 is a schematic block diagram of an electronic system 200 in accordance with another embodiment of the present invention. In the electronic system 200, at least one of the chips 102a and 102c (Figure 1) may be a memory device or the like coupled to a bus 204 that may be referred to as a processor bus. The processor bus 204 may be coupled to a direct memory access (DMA) controller 208. Direct memory access permits writing or reading directly to the memory chip 102a. The DMA controller 208 may be coupled to the first communications interface 112 by the internal bus 110 that may be referred to as a peripheral bus. Similarly, the second communications interface 114 may be coupled to a second DMA controller 214 by the peripheral or internal bus 120, and the second DMA controller 214 may be coupled to the chip 102b that may be a processor or the like. The chip 102b is coupled to the DMA controller 214 by a processor bus 220. The electronic system 200 may include at least a second processor chip 102d or additional devices coupled to the DMA controller 214 by the processor bus 220. The communications interfaces 112 and 114 may be communications interfaces for communications between any types of chips or may be BB-MM interfaces.

Figure 3 is a schematic block diagram of the first communications interface 112 in accordance with an embodiment of the present invention. The second communications interface 114 may be identical to the first communications interface

112. The first communications interface 112 may include a bus interface 300 coupled to the internal bus 110. The bus interface 300 includes a plurality of transmit control registers 302 and a plurality of receive control registers 304. The plurality of transmit control registers 302 are coupled to a plurality of transmit channels 306. The plurality of transmit channels 306 may also include one or more virtual general purpose input/output (GPIO) channels 307. The plurality of transmit channels 306 and the virtual GPIO channel 307 are coupled to a transmit (TX) control block 308. The outputs of the transmit control block 308 are the outbound links or pins 116. The receive control registers 304 are coupled to a plurality of receive channels 310. The receive channels 310 are coupled to a receive (RX) control block 312. Each of the transmit channels 306 may include a transmit first in first out (FIFO) 314 type buffer or memory device and each of the receive channels 310 may include a receive FIFO 316. The GPIO channel 307 may also include a first in first out (FIFO) 314 type buffer or memory device.

In accordance with an embodiment of the present invention, the transmit control block 308 may include a link controller 318 and a channel arbiter 320. The channel arbiter 320 determines which of the plurality of transmit channels 306 is to be activated or selected next to transmit data. As described in more detail below this could be one of a plurality of data channels, a virtual general-purpose input/output (GPIO) channel 307 or a message flow control (MFC) channel. The channel arbiter 320 uses as inputs the transmit channels 306, if any, that may be in a "wait" state and therefore cannot transmit data for some reason, the channel number of the currently activated transmit channel 306, and information from each of the transmit channels 306 or transmit FIFOs 314 indicating if they contain any data to be sent. The channel arbiter 320 outputs the channel number of the next transmit channel 306 or FIFO 314 to be activated for transmitting data.

The link controller 318 sends data from the active transmit channel 306 or FIFO 314 or from the virtual GPIO channel 307 across a selected one of the outbound links or pins 116. When the outbound link 116 is idle or the link controller 318 finishes sending a block of data across the selected outbound link 116, the link controller 318 uses the

channel number generated by the channel arbiter 320 to determine which of the plurality of transmit channels 306 to switch to or select next. After switching to a new transmit channel 306, the link controller 318 will again send data across the selected one of the outbound links or pins 116. It should be noted that the link controller 318 and channel arbiter 320 may be implemented in software.

The receive control block 312 may include a state machine 322 that stores the channel number of the currently active receive channel 306 or FIFO 316, the number of data bits in the current byte that have been transmitted and the data bits themselves in the current byte that have already been received. Using this information the state machine 322 will write each byte to the correct receive channel 310 or receive FIFO 316 after the state machine 322 has received a complete byte of data. The state machine 322 may also be implemented in software.

In accordance with an embodiment of the present invention, the communications interface 112 may include a power management unit 324. The power management unit 324 may be contained within the bus interface 300 or may be external to the bus interface 300. The power management unit 324 may be coupled to the plurality of transmit channels 306, plurality of receive channels 310, the transmit and receive control block 308 and 312, and the semiconductor chips 102 (Figures 1 and 2). The power management unit 324 facilitates placing the components of the system 200 into an idle state or sleep state or mode to conserve energy as described in more detail below. Before entering a sleep mode, the components may perform software handshaking. For example, the components may cause a sleep request message to be sent, receive an okay response and then enter the sleep mode. Sending and receiving these messages can be implemented by using any channel 306 or 307 and an associated control registers 302. Before sending a request-to-sleep message, the requesting chips 102 should transmit all of its data or messages in any transmit channels 306 or FIFOs 314 to stop outbound activity. Before responding with the okay to sleep message, the receiving chip 102 should receive all messages directed to it or drain any receive channels 310 or FIFOs 316 containing messages for the receiving chip 102 and

terminate all receive activity. Once the requesting chips 102 receive the okay to sleep response, it can safely enter the sleep mode.

If the chip 102 needs to wake up, it may do so without notifying the other chips 102. However, it is recommended that the waking chip send a message via any channel 306 or 307 signifying that it is waking up.

Figure 4 is a schematic block diagram of the communications interface 112 illustrating examples of a transmit control block 308 and a receive control block 312 in accordance with another embodiment of the present invention. The transmit control block 308 may include a multiplexer or mux 400 coupled to the plurality of transmit channels 306. The multiplexer 400 is coupled to a parallel in serial out (PISO) converter 402. The PISO converter 402 is coupled to a channel selector 404 and a control logic circuit 406. The channel selector 404 and the control logic circuit 406 are each coupled to a channel register 408 that is also coupled to the PISO converter 402. The channel selector 404 and the control logic circuit 406 are also connected to a channel configuration register 410 and a channel status register 412. The channel configuration register 410 and the channel status register 412 may be included as part of the transmit control registers 302 (Figure 3) and contained in the bus interface 300. A channel configuration register 410 may be associated with each transmit channel 306 and each receive channel 310 and provides information about the specific transmit channel 306 or receive channel 310 such as the type of service selected (DMA, interrupt), the threshold level, the type message flow control and the like. A channel status register 412 may also associated with each transmit channel 306 and each receive channel 310 and provides information about the channel 306 or 310 such as whether the channel 306 or 310 or FIFO 314 or 316 is in a "Wait" state, is empty or full and the degree or amount of fullness or emptiness or if there is any data in the FIFO 314 or 316.

In operation, the channel selector 404 determines which transmit channel 306 is to be selected or activated next. The selected transmit channel 306 may be a data channel, the virtual GPIO channel 307 or a message flow control (MFC) channel as described in more detail below. The channel selector 404 utilizes as inputs configuration data from the channel configuration register 410 and status data from the

channel status register 412 to determine or select the next transmit channel 306 to be activated to send data or information. The channel selector 404 provides as an output the channel number of the next transmit channel 306 and corresponding receive channel 310 to be selected or activated. The channel selector 404 determines which of the transmit channels 306 are ready to be activated in response to the configuration data and status data from the channel configuration register 410 and the channel status register 412. For example, transmit channels 306 with no data in their FIFOs 314 or in a wait state are not ready to be activated. A predetermined algorithm may be used to determine the next transmit channel 306 to be activated to transmit the data contained in its FIFO 314. Any algorithm can be used to select the next transmit channel 306 from among the ready transmit channels 306. For example, a round robin type algorithm or selection process may be used.

The control logic circuit 406 determines when a new transmit channel 306 is to be activated. A strobe (STB) signal 414 is generated by the control logic circuit 406 and transmitted by an outbound strobe link 415 to the other receiving or target communications interface 114 (Figure 2) when the control logic circuit 406 determines that a new channel 306 is to be activated or selected. The STB signal 414 also causes the new transmit channel number from the channel selector 404 to be stored by the channel register 408. The channel register 408 communicates the new transmit channel number to the PISO converter 402, the control logic circuit 406 and the multiplexer 400. The PISO converter 402 also sends the new channel number to the other receiving or target communications interface 114 over an outbound link or pin 116 or data (DAT) link.

Data signals or messages from a currently selected or activated one of the plurality of transmit channels 306 or FIFOs 314 is multiplexed by the multiplexer 400 using the currently activated channel number as the select bit for the multiplexer 400. The data is transferred by the multiplexer 400 to the PISO converter 402 or the data is read by the PISO converter 402 along with the channel number of the transmit channel 306 to be activated from the channel register 408 in response to the STB signal 414 being received by the channel register 408 and the PISO converter 402. The PISO

converter 402 converts the parallel data read from the transmit FIFO 314 to a serial bit stream or data signals (DAT) 413 that is sent across the outbound links or pins 116. The outbound links or pins 116 correspond to the inbound links or pins 118 of the receiving or target communications interface 114. The PISO converter 402 also serializes and transmits the new channel number being activated with the data stream DAT 413.

The receive control block 312 includes a serial in parallel out (SIPO) converter 416 that is coupled to the inbound links or pins 118. The SIPO converter 416 is coupled to a demultiplexer 418 and a channel register 420. The channel register 420 is coupled to the demultiplexer 418 and to a control logic circuit 422. The demultiplexer 418 is coupled to the plurality of receive channels 310, and the control logic circuit 422 is coupled to another channel configuration register 410 and another channel status register 412 that may be included as part of the receive control registers 304 and included in the bus interface 300. In operation, the SIPO converter 416 continually converts data stream DAT signals 413 on the inbound links or pins 118 from serial data to parallel data. This data is transferred to both the channel register 420 and the demultiplexer 418. The channel register 420 stores the parallelized inbound data in response to the STB signal 414 on the strobe link 415. The STB signal 414 indicates that a new transmit channel 306 and a new corresponding receive channel 310 are being selected or activated. The control logic circuit 422 also receives the STB signal 414 and the channel number of the new receive channel 310 to be selected.

The receive control logic circuit 422 may transmit a "Wait" signal 423 over a "Wait" link 424 to the transmit control logic circuit 406 of the transmit control block 308 if the configuration and status data from the receive channel configuration register 410 and the receive channel status register 412 indicate that the new receive channel 310 or FIFO 316 is full, disabled or otherwise cannot receive data. The control logic circuit 422 will also generate and send a "WRITE STROBE" signal 425 to the demultiplexer 418 when a whole or complete byte of data has been received by the SIPO converter 416. The demultiplexer 418 selects the proper receive channel 310 in response to the channel number from the channel register 420. Parallelized data from

the SIPO converter 416 will then be written into the selected one of the receive FIFOs 316 in response to the "WRITE STROBE" signal 425.

Figure 5 is a schematic block diagram of the communications interface 112 showing examples of control registers 302 (Figure 3) that may be used to transmit signals or data to other chips 102 or devices in accordance with an embodiment of the present invention. The control registers 302 are shown as forming at least a portion of the bus interface 300; although, the control registers 302 could be located outside or independently of the bus interface 300. The control registers 302 may include an interface mode register 502, a transmit FIFO register 504 associated with each channel 306, a channel status register 412 associated with each channel 306, as previously discussed with respect to Figure 4, an end of message (EOM) register 508 associated with each channel 306, a channel configuration register 410 associated with each channel 306, as previously discussed with respect to Figure 4, an interface interrupt identification (ID) register 512, a transmit frequency select register 514, a wait count register 516, a clock stop time register 518, and an interface width register 520.

The interface mode register 502 may be coupled to the chip 102 which may be a processor chip memory device or other device. The communications interface 112 may be operated in different modes, such as a standard mode, a legacy mode or other modes. The interface mode register 502 controls in which mode the communications interface 112 will operate.

The transmit FIFO register 504 writes data from the chip 102 to the associated transmit channel 306 or FIFO 314. The transmit FIFO register 504 may be accessed after the transmit channel 306 or FIFO 314 drops below a threshold value defined in the channel configuration register 410. The transmit FIFO register 504 may be accessed either directly by the processor, memory or other type chip 102 after an interrupt or polling signal or via direct memory access (DMA). If there is a multiple byte message transfer, the bytes may be placed in little-endian order or with the least significant digit first.

The channel status register 412 is coupled to the transmit channel 306 and to the transmit control block 308. Figure 6 shows an example of a bit layout and bit

definitions for the channel status register 412. The channel status register bit layout shown in Figure 6 may be used for either a transmit channel 306 or a receive channel' 310. The channel status register 412 may include information about whether the receive channel 310 received an end of message (EOM) signal 602; whether the receive channel
5 or FIFO 310 or the transmit channel or FIFO 306 is in a wait state; whether the receive FIFO 310 or the transmit FIFO 306 is full or empty 606 and 608; and the number of bytes of data 610 in either the receive FIFO 310 or the transmit FIFO 306.

Referring back to Figure 5, the EOM register 508 is coupled to a corresponding transmit channel 306 or FIFO 314 and indicates that an entire or complete message has
10 been written to the corresponding transmit FIFO 314.

As previously discussed, the channel configuration register 410 is coupled to the transmit channel 306 and to the transmit control block 308. Figure 7 shows an example of a bit layout and bit definitions for the channel configuration register 410. The channel configuration status register bit layout shown in Figure 7 may be used for either
15 a transmit channel 306 or a receive channel 310. The channel configuration register 410 may include information about whether early end of descriptor chain (EOC) service is selected 702, that is, whether a message can be interrupted before the end of the message is read; the type of receive or transmit FIFO service selected, DMA or interrupt
20 704; the receive or transmit FIFO service threshold 706; transmit and receive flow control, direct flow control (DFC) or message flow control (MFC) 708 and 710; and whether an associated receive FIFO 316 or an associated transmit FIFO 314 is enabled to receive or transmit messages or data 712.

Referring back to Figure 5, the interface interrupt ID register 512 may be coupled to the chip 102 and to the transmit control block 308. Figure 8 shows an
25 example of a bit layout and bit definitions for the interface interrupt ID register 512. Interrupts can be generated when a transmit FIFO 314 or a receive FIFO 316 reaches its threshold value set by its corresponding channel configuration registers 410, when an end of message (EOM) is receive, or when a DMA descriptor chain is reached before the end of a message is read. Generating an interrupt on an early DMA end of channel
30 is necessary to inform a processor type chip 102 of improper DMA programming. As

shown in Figure 8, each interface interrupt type has a bit associated with it in the interface interrupt ID register 512. When an interface interrupt occurs, the corresponding bit is set in the interface interrupt ID register 512.

Referring back to Figure 5, the transmit frequency select register 514, the wait count register 516, the clock stop time register 518 and the interface width register 520 may each be coupled between the chip 102 and the transmit control block 308. The transmit frequency select register 514 selects the clock speed of the outbound link 116. The wait count register 516 determines the time (in transmit clock cycles) that the transmit control block 308 will wait before retrying a transmit to a receive channel 310 that sent a wait signal 423 (Figure 4). Each transmit channel 306 has an independent wait count register 516 that counts the time after a wait signal 423 is received before a retransmission to the receive channel 310 that caused the wait signal 423 to be sent. The clock stop time register 518 determines the time (in transmit clock cycles) that a clock signal will stop transitioning after the outbound link 116 become idle. It should be noted that a clock signal may be generated by the interfaces 112 and 114 of the present invention only when needed. The interface width register 520 specifies the width or number of data links 116 that the first communications interface 112 will transmit over simultaneously to the second communications interface 114.

Figure 9 is a schematic block diagram of the first communications interface 112 illustrating examples of control registers 304 that may be used to receive data from the second communications interface 114 in accordance with an embodiment of the present invention. The control registers 304 are shown in Figure 9 as part of the bus interface 300 but could be located separate from the bus interface 300. The control registers 304 may include an interface mode register 502, a receive FIFO register 600 associated with each receive channel 310, a channel status register 412 associated with each receive channel 310, an end of message (EOM) register 508 associated with each receive channel 310, a channel start threshold register 602 associated with each receive channel 310, a channel stop threshold register 604 associated with each receive channel 310, an interface interrupt ID register 512, a wake-up register 606, a channel configuration register 410 associated with each receive channel 310 and an interface width register

520. The interface mode register 502, the channel status register 412, the EOM register 508, the interface interrupt ID register 512, the channel configuration register 410 and the interface width register 520 are the same or similar registers to those previously described with respect to Figures 4 and 5.

5 The receive FIFO register 600 is coupled between an associated receive channel 310 or FIFO 316 and the chip 102 to receive data. The receive FIFO register 600 reads data from the associated one of the receive channels 310 or FIFOs 316. When a multiple byte transfer occurs, the bytes may be placed in little-endian order. When the receive data is read from the receive FIFO 316 by the associated receive FIFO register 600, the data may be removed from the receive FIFO 316. Generally, the receive FIFO register 600 may be accessed after the receive FIFO 316 exceeds its threshold value as defined in the channel configuration register 410 or when an EOM message or signal is receive by the interface 112. The receive FIFO register 600 may be accessed directly by the chip 102 after an interrupt or polling signal or the FIFO register 600 may be accessed via DMA.

15 The channel start threshold register 602 and the channel stop threshold register 604 are each coupled between an associated one of the receive channels or FIFOs 316 and the chip 102 receiving data. The channel start threshold register 602 and the channel stop threshold register 604 store the respective start and stop threshold values for the associated receive FIFO 316. When the amount of data or bits in a receive FIFO 316 exceeds the stored value in the channel stop threshold register 604, a stop message for the associated receive FIFO 316 is sent to the source communications interface 112 sending the data to place the associated transmit FIFO 314 in a wait state. When the amount of data or bits in the receive FIFO 316 falls below the value in the channel start threshold register 602, a start message is sent for that receive FIFO 316 to the source communications interface 112 sending the data and the associated transmit FIFO 314 is reactivated or taken out of the wait state to continue transmitting the data or message. The threshold value stored in the channel stop threshold register 604 should be higher than the threshold value stored by the channel start threshold register 602 for the interface 112 to function properly.

The wake-up register 606 is coupled between the receive channel 310 or FIFO 316 and the chip 102. The wake-up register 606 is used to wake up the connected chip 102.

Figure 10 is a schematic block diagram of a communications interface 112 illustrating examples of control registers 302 that may be used to send and receive general purpose input/output (GPIO) signals 1000 or data in accordance with an embodiment of the present invention. The control registers 302 for performing virtual GPIO functions may include a virtual GPIO input and output pin-level register 1001 associated with each virtual GPIO channel 307, a virtual GPIO output pin-set register 1002 and a virtual GPIO pin-clear register 1004 both associated with each virtual GPIO channel 307, a virtual GPIO rising and falling edge detect register 1006 associated with each GPIO channel 307, a virtual GPIO edge detect status register 1008 associated with each virtual GPIO channel 307, and a virtual GPIO value interrupt register 1010. The virtual GPIO input and output pin-level register 1001 is coupled between the virtual GPIO channel 307 and the chip 102 and provides the state or status of each GPIO pin 1012 for sending GPIO data. The virtual GPIO output pin-set and pin-clear registers 1002 control the state on each GPIO pin 1012. The virtual output GPIO pin 1012 is set by writing a 1 to the corresponding virtual GPIO output pin-set register 1002 and the virtual GPIO output pin 1012 is cleared by writing a 1 to the corresponding virtual GPIO pin-clear register 1002. The virtual GPIO rising and falling edge register 1006 is coupled between the virtual GPIO channel 307 and the chip 102. The virtual GPIO rising and falling edge register 1008 configures the GPIO pin 1012 to detect either a rising-edge transition, a falling edge transition or both. When such a transition is detected, a bit is set in the virtual GPIO detect status register 1008. The virtual GPIO value interrupt register 1010 is coupled between the virtual GPIO channel 307 and the chip 102. The virtual GPIO value interrupt register 1010 contains a configuration bit that may be set to specify if an interrupt is to be generated when a virtual GPIO value or signal is received by the first communications interface 112 across an inbound link 118 (Figure 1).

Figure 11 is a block schematic diagram of an example of the first communications interface 112 and the second communications interface 114 coupled by different outbound communications links or pins 116 and inbound communications links or pins 118 and examples of signals that may be transmitted over each of the links 116 and 118 between the first communication interface 112 and the second communications interface 114. The outbound links or pins 116 may include a clock link or pin 1102 to send a CLK signal, a strobe link or pin 1104 to send a STB signal, a wait link or pin 1106 to send a WAIT signal and a plurality of data links or pins 1108 that are used to facilitate the transmission of data, DAT signals, or messages from the first interface 112 to the second interface 114. Similarly, the inbound links or pins 118 also may include a clock link 1110 to send a CLK signal, a strobe link 1112 to send a STB signal, a wait link 1114 to send a WAIT signal and a plurality of data links or pins 1116 to facilitate the transmission of data, DAT signals, or messages from the second interface 114 to the first interface 112. An interface 112 or 114 sending or transmitting data may be referred to as a source or a source interface and an interface 112 or 114 receiving data may be referred to as a target or a target interface.

In accordance with an embodiment of the present invention, an example of channel number assignments or designations for the transmit and receive channels 306 and 310 (Figure 3) and a description of the function of each channel 306 and 310 according to its number assignment is shown in Figure 12. Channel 0 may be a null channel to send an end of message (EOM) signal as previously discussed. Channels 1-7 may be used for the transmission of data or messages. If a source communications interface 112 must stop transmitting a message without activating a new channel 1-7, the communications interface 112 may activate channel 11, the empty channel. Channel 11 may also be designated as channel B in hexadecimal notation. Channel 13 or D may be used as the virtual GPIO channel 307 (Figure 3). Channel 14 or E and Channel 15 or F may be used to send stop and start messages if the transmission of data to a selected receive FIFO 316 must be halted for some reason.

Figure 13 is an example of signal waveforms to transmit the message "7B3D" over data channel 3. The first waveform is the clock or CLK signal 1302 and is

transmitted on the clock link or pin 1102 (Figure 11). The second waveform is the data or DAT signal 1304 and is transmitted on the data links 1108 (Figure 11). The third waveform is the strobe or STB signal 1306 and is transmitted on the strobe link or pin 1104 (Figure 11). The fourth waveform is the wait signal 1308 and is transmitted on the wait link or pin 1106 (Figure 11). One of the data channels 1-7 (Figure 12) is activated by generating a STB signal or pulse 1310 on the strobe link 1104 or pin and indicating the data channel number 1-7, in this case channel number 3 illustrated by pulse 1312 in Figure 13, on a corresponding data link (DAT) 1108 before the next CLK signal or pulse 1314. If the communications interface 112 is set to detect rising edges as opposed to falling edges of the CLK signal 1302, channel 3 will be selected or will become the active channel 306 and 310 for transmitting and receiving data when the next rising edge transition of the CLK signal 1314 is detected. The data signals or message "3D7B" 1304 will be transferred over channel 3 on each of the following rising-edge transitions of the CLK signal 1302.

As previously mentioned, the communications interfaces 112 and 114 may support different interface widths. The appropriate bit in the interface width register 520 may be set to provide the different interface widths, such as a serial width or mode (1 bit), a two-bit width or mode, a nibble width or mode (4-bits) and so forth. In the example shown in Figure 13 the signals 1302, 1304, 1306 and 1308 are in the nibble width or mode. Accordingly, four data links or pins 1108 in Figure 11, DAT(0), DAT(1), DAT(2) and DAT(3), may be used to transmit a message in nibble mode.

When the transmission of the message through a data channel 1-7 (Figure 12) is complete, the data link (DAT) 1108 (Figure 11) may change the active channel designation to channel 0, or the null channel (Figure 12). Switching to the null channel signifies the end of a message (EOM) and therefore initiates service at the target communications interface 114 for the corresponding receive FIFO 316 or channel 310 to become active. An end of message (EOM) signal or pulse 1316 is shown at the end of the data bit stream or signal 1304 in Figure 13.

Activating a new data channel 1-7 requires reassertion of the STB signal 1306 on the strobe link or pin 1104 (Figure 11) and transmitting the new data channel number

1-7 on the data links 1108. The new data channel 1-7 may be activated when no data transfers are occurring , in the middle of a data transfer on another data channel 1-7 or just after the current data transfer has finished and the null channel 0 has been activated to indicate the end of a message. An example of selecting or activating a new data channel 1-7 is illustrated in Figure 14. In the example of Figure 14, channel 3 has been activated by the STB signal or pulse 1402 and sending a channel number "3" pulse 1404 on the data link 1108 to send the message "3D7B." After transmitting the "7B" signal 1406, another STB signal or pulse 1408 is generated and a data signal 1410 designating channel number 2 is sent on the data link 1108 to activate channel 2 to send the message "AE" 1412. After the end of message (EOM) signal 1414, channel 3 is again activated by an STB signal or pulse 1416 and a channel number "3" signal 1418 to reactivate channel number 3 and send the remainder of the message "3D" 1420. The message on channel 3 may be pre-empted by the message on channel 2 because the channel 2 message may have a higher priority.

While receiving data, the receive FIFO 316 (Figure 3) can become full which would prevent the receive FIFO 316 from accepting new data. One example of a flow control method to notify the source communications interface 112 of this condition may be referred to as direct flow control (DFC) and another example of a flow control method for a FIFO full condition may be referred to as message flow control (MFC). Both methods temporarily disable data transfers by putting the active transmit channel 306 or FIFO 314 in a "wait" state. When the active transmit channel 306 is in a wait state, the source communications interface 112 cannot send any data through that channel 306. Any attempt to transmit data will be ignored. Either or both flow control methods can be used by the communications interfaces 112 and 114.

Referring back to Figure 11, in the direct flow control method, the target interface 114 will assert a "wait" signal 1308 over the wait link or pin 1106 to the source interface 112, if the active receive channel 310 or FIFO 316 is disabled, invalid or full. The "wait" signal 1308 will also be sent after a reset and while the data link 1108 is idle, i.e., there is no data or messages being transmitted. The source interface 112 will sample the "wait" signal 1308 on each CLK pulse 1302 of the CLK link 1102

while the active data channel 1-7 is in a wait state for as long as the "wait" signal 1308 is being asserted. When the "wait" signal 1308 goes low or is no longer asserted, data transmission can resume. Another data channel 1-7 may be activated while the currently active channel 1-7 is in a wait state by transmitting the STB signal 1306 on the strobe link 1104 and transmitting a new data channel number 1-7 on a corresponding data link 1108.

Figure 15 is an example of message flow control (MFC) that uses stop and start messages 1502 and 1504 to put an active data channel 1-7 in a wait state. When a receive FIFO 316 exceeds a user-programmable threshold level set in the channel stop threshold register 604, the corresponding channel 1-7 will be placed in a wait state by sending a stop message 1502. The stop message 1502 is sent by transmitting the channel number, for example channel 4, on channel 14 or channel E in hexadecimal on a data link 1116 (Figure 11) from the target interface 114 to the source interface 112. Accordingly, a strobe signal or pulse 1506 is sent along with the stop channel number E 1502 to activate the stop channel 14 or E (Figure 12). The number of the data channel 1-7 to be placed in the wait state is then transmitted on the stop channel E. In the example of Figure 15, a signal 1508 designating channel 4 is transmitted on channel E or the stop channel. When the source interface 112 receives the stop message 1502, the active data channel 1-7 will enter a wait state and stop sending data until taken out of the wait state. The channel 1-7 exits the wait state when the source interface 112 receives a start message 1504. When the receive FIFO 316 for the active channel 1-7 drops below a user selected threshold level set in the channel start threshold register 602, a start message 1504 is sent by transmitting the channel number of the channel 1-7 to be reactivated over channel 15 or channel F in hexadecimal on an outbound data link 1116 from the target interface 114 to the source interface 112. Accordingly, in the example of Figure 15, another strobe signal or pulse 1510 is transmitted on the strobe link 1104 and the start channel designation "F" signal 1504 is transmitted on the data link 1116 followed by the channel number "4" signal 1512 to take channel 4 out of the wait state. As previously discussed, the threshold levels for sending the channel stop and start messages 1502 and 1504 may be set by a user in the channel stop threshold

register 604 and the channel start threshold register 602 (Figure 9). Message flow control has higher priority than other messages being sent on the data links 1108 and 1116 and will preempt other message traffic as soon as the current byte is sent.

Figure 16 is a flowchart of an example of a method 1600 of transmitting data or messages between semiconductor chips 102 or other devices in accordance with an embodiment of the present invention. In block 1602, data is written into at least one of the plurality of transmit FIFOs 314. In block 1604, one of the plurality of transmit FIFOs 314 that contains data and is not in a wait state according to a predetermined algorithm, such as round-robin or the like, is selected to form an active channel 1-7 for transmitting the data from the source interface 112 to the target interface 114. In block 1606 a strobe signal is sent from the source interface 112 to the target interface 114 to initiate the transmission of data. In block 1608, the selected channel number of the active channel 1-7 is transmitted over a selected data link 1108 from the source interface 112 to the target interface 114. In block 1610, a corresponding one of a plurality of receive FIFOs 316 that is not full or in a wait state is selected to form the active channel 1-7 and in block 1612, the data or message is transmitted over a corresponding data link 1108 from the active channel 1-7 transmit FIFO 314 of the source interface 112 to the corresponding one of the receive FIFOs 316 of the target interface 114. In block 1614, an end of message (EOM) signal 1316 is sent after all of the data has been transmitted. In block 1616, a wait signal 1308 or a stop message 1502 may be sent from the target interface 114 to the receive interface 112 if the corresponding one of the receive FIFOs 316 cannot receive data because it is disabled, invalid, full or for some other reason cannot receive data. In block 1618, the wait signal 1308 is removed or a start message 1504 may be sent if the corresponding one of the receive FIFOs 316 can now receive data. In block 1620, at least one other transmit FIFO 314 and another corresponding receive FIFO 316 may be selected or activated to form an active channel 1-7 while the one receive FIFO 316 cannot receive data.

Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiment

[illegible]